Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Theoretical Computer Science 306 (2003) 543–551

Theoretical
Computer Science

www.elsevier.com/locate/tcs

Note

An asymptotic fully polynomial time approximation scheme for bin covering

Klaus Jansen[a,1], Roberto Solis-Oba[b,*,2]

[a] *Institut für Informatik und Praktische Mathematik, Universität zu Kiel, Germany*
[b] *Department of Computer Science, University of Western Ontario, London ONT. N6A 5B7, Canada*

Received 17 July 2002; received in revised form 20 February 2003; accepted 16 June 2003
Communicated by G.F. Italiano

Abstract

In the bin covering problem there is a group $L=(a_1,\ldots,a_n)$ of items with sizes $\tilde{s}(a_i)\in(0,1)$, and the goal is to find a packing of the items into bins to maximize the number of bins that receive items of total size at least 1. This is a dual problem to the classical bin packing problem. In this paper we present the first asymptotic *fully polynomial-time* approximation scheme for the problem.

Keywords: Asymptotic approximation scheme; Bin covering; Linear programming

1. Introduction

Given a group $L=(a_1,\ldots,a_n)$ of items with sizes $\tilde{s}(a_i)\in(0,1)$, the bin covering problem is to find a packing of the items into bins to maximize the number of bins that receive items of total size at least 1. This problem is considered to be a kind of dual to the classical bin packing problem, and sometimes it is called the dual bin packing problem.

The bin covering problem is NP-hard and, furthermore, it cannot be approximated within a factor $\alpha>\frac{1}{2}$ of the optimum, unless P = NP [1]. On the positive side, the

* Corresponding author. Tel.: 519-6612111-86974; fax: 519-6613515.
E-mail addresses: kj@informatik.uni-kiel.de (K. Jansen), solis@csd.uwo.ca (R. Solis-Oba).

[1] Supported in part by EU Project APPOL I + II, Approximation and Online Algorithms, IST-1999-14084 and IST-2001-30012.
[2] Supported in part by the Natural Sciences and Engineering Research Council of Canada grant R3050A01.

0304-3975/03/$ - see front matter
doi:10.1016/S0304-3975(03)00363-3

next-fit algorithm achieves a performance ratio of $\frac{1}{2}$ [2]. Interestingly, if the number of bins filled by an optimum solution is large, then algorithms with performance ratio better than $\frac{1}{2}$ can be designed. Given a group L of items and an algorithm A, let $A(L)$ be the number of bins that A can fill with items of total size at least 1. Let $OPT(L)$ be the numbers of bins covered by an optimal algorithm. The *asymptotic worst case ratio* of algorithm A is defined as

$$R_{\infty}^{A} = \liminf_{OPT(L)\to\infty} \frac{A(L)}{OPT(L)}.$$

Algorithms for the bin covering problem with asymptotic worst case ratios $\frac{2}{3}$ and $\frac{3}{4}$ are given in [1–3]. Recently, an algorithm with asymptotic ratio $1-\varepsilon$, for any $\varepsilon>0$, was designed by Csirik et al. [4]. This algorithm runs in time polynomial in n but exponential in $1/\varepsilon$.

In this paper we present an asymptotic fully polynomial time approximation scheme (AFPTAS) for the bin covering problem, i.e., an algorithm with asymptotic worst case ratio $1-\varepsilon$, $\varepsilon>0$, and running time polynomial in both n and $1/\varepsilon$. This algorithm narrows the gap between the approximability of the bin packing problem and the bin covering problem, and it solves the question first posed in [2] of whether an AFPTAS for the bin covering problem exists.

Our algorithm is a faster version of the algorithm in [4] that does not have the exponential dependency of the time complexity on $1/\varepsilon$. The approach in [4] is to formulate the problem as an integer program (IP), and then round a basic optimum solution of a certain linear program relaxation of IP. This linear program has a number of constraints that is exponential in $1/\varepsilon$, and so its solution needs time that is also exponential in $1/\varepsilon$.

We get a faster algorithm by using the potential price directive decomposition method of Grigoriadis and Khachiyan [5,6] to find a near optimum solution for the above linear program. The application of this method to our problem requires the efficient solution of two variants of the knapsack problem. We show that one of these variants is hard to approximate within a factor $1+\varepsilon$ of the optimum for small values $\varepsilon>0$. Despite this, we can still find an approximate solution for the linear program in time polynomial in n and $1/\varepsilon$. We present an efficient algorithm for transforming the near optimum solution for the linear program into a basic feasible solution, which is then rounded as in [4] to yield our asymptotic fully polynomial time approximation scheme.

2. An asymptotic polynomial time approximation scheme

In this section we briefly review the asymptotic polynomial time approximation scheme described in [4]. For any set $T\subseteq L$, let $\tilde{s}(T)=\sum_{a_i\in T}\tilde{s}(a_i)$. Clearly, $\tilde{s}(L)\geqslant OPT(L)$. For convenience, we assume that $\tilde{s}(L)\geqslant 2$, $\varepsilon<\frac{1}{2}$, and $1/\varepsilon$ is integer.

Partition the set L into three classes, $\mathscr{L}$, $\mathscr{M}$, and $\mathscr{S}$, respectively, formed by *large*, *medium*, and *small* items: (a) if $n<\lfloor\tilde{s}(L)\rfloor(1+1/\varepsilon)$ then $\mathscr{L}=L$, $\mathscr{M}=\mathscr{S}=\emptyset$;

(b) otherwise, place in $\mathscr{L}$ the largest $\lfloor \tilde{s}(L) \rfloor/\varepsilon$ items, put in $\mathscr{M}$ the next largest $\lfloor \tilde{s}(L) \rfloor$ items, and in $\mathscr{S}$ the remaining ones. Now, we need to consider two cases: $\tilde{s}(L) > 13/\varepsilon^3$ and $\tilde{s}(L) \leqslant 13/\varepsilon^3$.

Case 1. $\tilde{s}(L) > 13/\varepsilon^3$. Sort the large items non-increasingly by size. Then, partition these items into $1/\varepsilon^2$ groups $G_1, G_2, \ldots, G_{1/\varepsilon^2}$ such that each group G_i, $1 \leqslant i \leqslant |\mathscr{L}| \bmod (1/\varepsilon^2)$ has $\lceil |\mathscr{L}|\varepsilon^2 \rceil$ items, and each one of the remaining groups has $\lfloor |\mathscr{L}|\varepsilon^2 \rfloor$ items. The largest size items are placed in G_1, the next largest size items are put in G_2, and so on. For each group G_i, round the sizes of the items down to the size of the smallest item in the group.

Let H be the set of different sizes after rounding. Note that $|H| \leqslant 1/\varepsilon^2$. For an item size $u \in H$, let $n(u)$ be the number of large items of rounded size u. Let $s(a_i)$ be the rounded size of item a_i. For small and medium items a_i, let $s(a_i) = \tilde{s}(a_i)$. Given a set of items $T \subseteq L$, let $s(T)$ be the sum of the rounded sizes of the items in T.

We define a *bin configuration* C as a set of large items of total rounded size smaller than 2. For a bin configuration C and size $u \in H$, let $n(u, C)$ be the number of large items of size u in C. Let ζ be the set of all possible bin configurations. The bin covering problem can be expressed as an integer program by using variables x_C that indicate the number of bins that contain items according to configuration C. If we relax the integrality constraint on the variables x_C and assume that small items can be split, we get the following linear program.

$$\begin{aligned} \text{Max} \sum_{C\in\zeta} x_C & \\ \text{s.t.} \sum_{C\in\zeta} n(v,C)x_C \leqslant n(v), & \quad \forall v \in H, \\ \sum_{\substack{C\in\zeta \\ s(C)<1}} (1 - s(C))x_C \leqslant s(\mathscr{S}), & \quad x_C \geqslant 0, \ \forall C \in \zeta. \end{aligned} \tag{1}$$

Let $(x_C^*)_{C\in\zeta}$ be a basic optimum solution for (1). Round each value x_C^* down to the nearest integer value $y_C^* = \lfloor x_C^* \rfloor$. The large items are placed in a set D of bins according to the rounded solution $(y_C^*)_{C\in\zeta}$. Next, the medium and small items are greedily placed into those bins in D that contain items with total size smaller than 1.

Case 2. $\tilde{s}(L) \leqslant 13/\varepsilon^3$. $\mathscr{L}$ contains at most a constant, $13(1 + 1/\varepsilon)/\varepsilon^3 < 20/\varepsilon^4$, number of items. Hence, exhaustive search can be used to place them into the maximum number of bins so that the total unused space is at most $\tilde{s}(\mathscr{S})$. Then, the medium and small items are placed as described before.

The above algorithm needs to find a basic optimum solution for linear program (1), and this requires time exponential in $1/\varepsilon$ since the linear program has $O(n^{1/\varepsilon^2})$ variables. In [4] it is proved that the algorithm fills at least $OPT(L)$ $(1 - 5\varepsilon) - 4$ bins, and thus, it is an asymptotic polynomial time approximation scheme.

3. Approximate solution of the linear program

In this section we describe an algorithm for approximately solving linear program (1) in time polynomial in n and $1/\varepsilon$. First, we modify the above definition of a bin configuration. If $\mathscr{S}=\emptyset$, then we define a bin configuration as a set of large items of total rounded size smaller than 2, but at least 1. Hence, if $\mathscr{S}=\emptyset$ the second constraint of linear program (1) disappears. If $\mathscr{S}\neq\emptyset$, then we use the same definition of bin configuration as above.

Note that if $\tilde{s}(L)\leqslant 13/\varepsilon^3$, then $OPT(L)\leqslant 13/\varepsilon^3$ is a constant. Hence, we can use the next-fit algorithm to fill at least $OPT(L)-O(1/\varepsilon^3)$ bins. Thus, for the rest of the paper we concentrate on the case when $\tilde{s}(L)>13/\varepsilon^3$.

Let X^* be the value of an optimum solution $(x_C^*)_{C\in\zeta}$ for linear program (1). Since $\tilde{s}(L)>13/\varepsilon^3$ and $\varepsilon<\frac{1}{2}$, then $X^*>1$. Also, every item has size at most 1, and so, $1<X^*\leqslant s(L)\leqslant n$. We can use the potential price directive decomposition method [5,6] to find a near optimal solution for linear program (1), but first, we need to re-write the linear program in a special form. Let us partition the interval $[1,n]$ into subintervals of size ε. Consider an endpoint $k\varepsilon$ of one of these subintervals. We re-write the linear program as follows:

$$\lambda^* = \min\left\{\lambda \;\middle|\; \sum_{C\in\zeta}\frac{n(v,C)}{n(v)}x_C \leqslant \lambda,\ \forall v\in H, (x_C)_{C\in\zeta}\in B_k,\ \text{and} \right.$$
$$\left. \sum_{C\in\zeta \text{ s.t. } s(C)<1}\frac{1-s(C)}{s(\mathscr{S})}x_C\leqslant\lambda,\ \forall (x_C)_{C\in\zeta}\in B_k\right\}, \tag{2}$$

where

$$B_k=\left\{(x_C)_{C\in\zeta}\;\middle|\;\sum_{C\in\zeta}x_C=k\varepsilon \quad\text{and}\quad x_C\geqslant 0 \text{ for all } C\in\zeta\right\}.$$

A feasible solution of value $\lambda^*=1$ for this new linear program (2) is a solution for the original linear program (1) of value $k\varepsilon$.

Linear program (2) is a *convex block-angular resource sharing problem* [5], and the price directive decomposition method [5,6] can be used to solve it to any given precision $\varepsilon>0$. This method needs to repeatedly solve the following *block problem*:

$$\min\{y^{\mathrm{T}}A(x_C)_{C\in\zeta}\mid (x_C)_{C\in\zeta}\in B_k\}, \tag{3}$$

where $y=(y_1,y_2,\ldots,y_{|H|+1})$ is a non-negative *price vector* and A is the constraint matrix corresponding to linear program (2).

In the next section we show how to solve this block problem. The price directive decomposition method can either (a) find a solution $(x_C)_{C\in\zeta}\in B_k$ such that $\sum_{C\in\zeta} n(v,C)/n(v)x_C\leqslant(1+\varepsilon)$ for every $v\in H$ and $\sum_{C\in\zeta \text{ s.t. } s(C)<1}(1-s(C))/s(\mathscr{S})x_C\leqslant(1+\varepsilon)$, or (b) it can conclude that there is no solution for which $\sum_{C\in\zeta} n(v,C)/n(v)x_C\leqslant 1$ for all $v\in H$ and $\sum_{C\in\zeta \text{ s.t. } s(C)<1}(1-s(C))/s(\mathscr{S})x_C\leqslant 1$. Thus, we can find an approximate value for the optimum solution X^* by performing a binary search over the ε-subintervals

on $[1,n]$ to find a value $k^*\varepsilon$ and a solution $(x'_C)_{C\in\zeta}$ with value $X^* \geqslant \sum_{C\in\zeta} x'_C = k^*\varepsilon \geqslant (1-\varepsilon)X^*$, and such that

$$\sum_{C\in\zeta} \frac{n(v,C)}{n(v)} x'_C \leqslant (1+\varepsilon), \qquad \forall v \in H$$

$$\sum_{C\in\zeta \text{ s.t. } s(C)<1} \frac{1-s(C)}{s(\mathscr{S})} x'_C \leqslant (1+\varepsilon).$$

To transform $(x'_C)_{C\in\zeta}$ into a feasible solution $(\tilde{x}_C)_{C\in\zeta}$ for linear program (1) we set $(\tilde{x}_C)_{C\in\zeta} = (x'_C(1-\varepsilon))_{C\in\zeta}$, so that

$$\sum_{C\in\zeta} \frac{n(v,C)}{n(v)} \tilde{x}_C \leqslant (1+\varepsilon)(1-\varepsilon) = 1-\varepsilon^2 \leqslant 1, \qquad \forall v \in H$$

$$\sum_{C\in\zeta \text{ s.t. } s(C)<1} \frac{1-s(C)}{s(\mathscr{S})} \tilde{x}_C \leqslant (1+\varepsilon)(1-\varepsilon) = 1-\varepsilon^2 \leqslant 1.$$

Furthermore, the value of the new solution $(\tilde{x}_C)_{C\in\zeta}$ is $\sum_{C\in\zeta} \tilde{x}_C \geqslant (1-\varepsilon)^2 X^* \geqslant (1-2\varepsilon)X^*$.

We notice that this solution $(\tilde{x}_C)_{C\in\zeta}$ might be formed by up to $O(|H|(1/\varepsilon^2 + \ln|H|)) = O(1/\varepsilon^4)$ configurations. But, since there are at most $1 + 1/\varepsilon^2$ linear constraints in the linear program, a basic feasible solution consists of at most $O(1/\varepsilon^2)$ configurations.

We transform $(\tilde{x}_C)_{C\in\zeta}$ into a feasible solution with the same value, but at most $1+1/\varepsilon^2$ positive components $\tilde{x}_C$ as follows. Consider a set Q of $2+1/\varepsilon^2$ configurations C with positive variables $\tilde{x}_C > 0$. Let M be the submatrix of A formed by the $2+1/\varepsilon^2$ columns corresponding to those variables. Solve the system $M(w_C)_{C\in Q} = 0$, where $(w_C)_{C\in Q}$ is a non-zero $(2+1/\varepsilon^2)$-dimensional vector. There exists always a non-zero solution of this system since matrix M is singular. Then, modify the solution by making $\tilde{x}_C = \tilde{x}_C + \delta w_C$ for all configurations $C \in Q$, where $|\delta|$ is the smallest value such that at least one of the new components $\tilde{x}_C + \delta w_C$ is zero. We repeat this process until we find a solution $(\tilde{x}_C)_{C\in\zeta}$ with at most $1 + 1/\varepsilon^2$ positive components.

3.1. The block problem

Consider block problem (3). Since B_k is a simplex, an optimum solution for (3) is attained at a vertex $(\bar{x}_C)_{C\in\zeta}$ of B_k corresponding to a single configuration C^*, i.e. $\bar{x}_{C^*} = k\varepsilon$ and $\bar{x}_C = 0$ for all other configurations $C \in \zeta$, $C \neq C^*$. Thus, to solve (3) it is sufficient to find a bin configuration C^* with smallest associated *price value* $y^{\mathrm{T}}A[C^*]$, where $A[C^*]$ is the column of A for configuration C^*.

Depending on the total size of the items in a configuration C, the configuration's price $y^{\mathrm{T}}A[C]$ can be either

- $\sum_{u\in C} n(s(u),C) y_{s(u)}/n(s(u))$ if $s(C) \geqslant 1$, or
- $\sum_{u\in C} n(s(u),C) y_{s(u)}/n(s(u)) + (1-s(C)) y_{|H|+1}/s(\mathscr{S})$ if $s(C) < 1$.

Therefore, configuration C^* can be found by solving the following two integer programs. Variable z_v denotes the number of items of size v chosen for the solution. In the first integer program we optimize over all configurations with size smaller than 1.

$$\tau_1 = \min \sum_{v\in H} \frac{y_v}{n(v)} z_v + \frac{y_{|H|+1}}{s(\mathscr{S})} \left(1 - \sum_{v\in H} s(v) z_v\right)$$
$$\text{s.t.} \sum_{v\in H} s(v) z_v \leqslant 1, \quad z_v \in \{0, 1, \ldots, n(v)\}. \tag{4}$$

In the second integer program, we optimize over all configurations $C \in \zeta$ with size $s(C) \geqslant 1$.

$$\tau_2 = \min \sum_{v\in H} \frac{y_v}{n(v)} z_v$$
$$\text{s.t.} \sum_{v\in H} s(v) z_v \geqslant 1, \quad z_v \in \{0, 1, \ldots, n(v)\}. \tag{5}$$

Configuration C^* has price value $y^{\mathrm{T}} A[C^*] = \min\{\tau_1, \tau_2\}$. The above integer Programs (4) and (5) are variations of the knapsack problem with bounded variables. Problem (5) is the minimization version of the knapsack problem, and it is not difficult to show that a simple dynamic programming algorithm solves it in polynomial time within a factor $(1+\varepsilon)$ of the optimum.

Problem (4), however, is more complicated to solve approximately. To see this, assume that the coefficients $y_v/n(v)$ are all equal to 0, and $y_{|H|+1} = 1$. Moreover, assume that the optimum solution consists of a set U of items of total size 1. Then, the value of the optimum solution is 0. A $(1+\varepsilon)$-approximation algorithm for the problem must find a solution of value 0 and, therefore, it must solve the subset sum problem.

Note, however, that we do not need to solve approximately both Problems (4) and (5) but, rather, we must find an approximate solution for problem

$$\tau^* = \min\{\tau_1, \tau_2\}. \tag{6}$$

3.1.1. Bounding the value of the optimum solution for the block problem

To solve Problem (6), let us first find lower and upper bounds for τ^*. Every item $u \in L$ has price $y_{s(u)}/n(s(u))$, and if the total size S' of the items placed in a bin is smaller than 1, then we might think that the empty part of the bin is filled with a dummy element of size exactly $1 - S'$ and price $(1 - S')y_{|H|+1}/s(\mathscr{S})$. It is convenient to think then, that besides the items in L there is an additional dummy item a_{n+1} of variable size (and price) that we can always use to completely fill a bin. Let L' be a list containing all elements from L plus element a_{n+1}. Then, problem (6) can be thought as that of selecting the minimum price set of elements from L' of total size at least 1.

Let $r(u) = [y_{s(u)}/n(s(u))]/s(u)$ for all items $u \in L$, and $r(a_{n+1}) = y_{|H|+1}/s(\mathscr{S})$. Sort the items $u \in L'$ in non-decreasing order of price-to-size ratio $r(u)$. Observe that we do not need to consider elements u with ratio $r(u)$ larger than or equal to $r(a_{n+1})$. Take

the items $u \in L'$ in this order and place them in a bin of size 1 until we find the first item w_1 that overflows the bin. (We consider that item a_{n+1} always overflows the bin.) Item w_1 is not placed in the bin. Let R_1 be the set of items that get placed into the bin. Let $s(R_1) \leqslant 1$ be the total size of the elements in R_1, and let $L^<(R_1)$ be the set of items not in R_1, each of size smaller than $1 - s(R_1)$. Similarly, let $L^>(R_1)$ be the set of items not in R_1 of size larger than $1 - s(R_1)$. Find the smallest price item p_1 in $L^>(R_1)$. Note that $R_1 \cup \{p_1\}$ is a feasible solution for (6).

Take the elements u in $L^<(R_1)$ in non-decreasing order of ratio $r(u)$ and add them to the bin containing R_1 until we find the first item w_2 that would overflow the bin. Let R_2 ($w_2 \notin R_2$) be the new set of items in the bin (including those from R_1), and let p_2 be the smallest price item in $L^>(R_2)$. We have now a second feasible solution $R_2 \cup \{p_2\}$. We repeat the above process with the elements in $L^<(R_2)$ to get a new feasible solution, and so on. Let $R_1 \cup \{p_1\}, R_2 \cup \{p_2\}, \ldots, R_k \cup \{p_k\}$ be the feasible solutions computed as described above. Notice that $p_k = a_{n+1}$. Let $R \cup \{p\}$ be the solution with minimum price in this group, and let

$$P = \frac{1}{2}\left[\sum_{u \in R} y_{s(u)}/n(s(u)) + y_{s(p)}/n(s(p))\right].$$

Lemma 1. $P \leqslant \tau^* \leqslant 2P$.

Proof. Let z^* be an optimum solution for Problem (6). If z^* does not contain any items from sets $L^>(R_i)$, $i = 1, \ldots, k-1$, then z^* consists of a subset of the items in $R_k \cup \{p_k\}$. Hence, because of the order in which the items are considered, the total price of $R_k \cup \{p_k\}$ is equal to the price of z^*. Therefore, $P \leqslant \sum_{u \in R_k} y_{s(u)}/n(s(u)) + y_{s(p_k)}/n(s(p_k)) = \tau^* \leqslant 2P$.

So, let us assume that z^* contains at least one item $u \in L^>(R_i)$, and let i be the least index for which this condition holds (so z^* does not contain any items from $L^>(R_j)$ for any $j < i$). Since we choose to put in the bin the items in non-decreasing order of price-to-size ratio, then the total price $\sum_{u \in R_i} y_{s(u)}/n(s(u))$, of the elements in R_i is not larger than τ^*. Moreover, since z^* contains at least one item from $L^>(R_i)$, and p_i is the smallest price item in $L^>(R_i)$, then the price $y_{s(p_i)}/n(s(p_i))$ of p_i is no larger than τ^*. Hence, $P \leqslant \frac{1}{2}(\sum_{u \in R_i} y_{s(u)}/n(s(u)) + y_{s(p_i)}/n(s(p_i))) \leqslant \tau^* \leqslant 2P$, where the last inequality follows since $R \cup \{p\}$ is a feasible solution of (6). □

3.1.2. Solving the block problem

We round up each one of the coefficients $y_v/n(v)$ in (4) and (5) to the nearest value of the form $i(\varepsilon/n)P$, where $i \in \{0, 1, \ldots, \lceil 2n/\varepsilon \rceil\}$. By Lemma 1, an optimum solution for problem (6) with rounded coefficients $y_v/n(v)$ has value at most $(1+\varepsilon)$ times the optimum solution for the problem with the original coefficients.

We can find an optimum solution for problem (6) with rounded coefficients by using dynamic programming as follows. Let the items in L be $a_1, a_2, \ldots, a_n$. Let $M(i, \ell)$ be the maximum size $s(T)$ of a subset $T \subseteq \{a_1, \ldots, a_i\}$ of price value $\sum_{a_i \in T} y_{s(a_i)}/n(s(a_i)) = \ell(\varepsilon/n)P$ for $i = 0, 1, \ldots, n$, $\ell = 0, 1, \ldots, \lceil 2n/\varepsilon \rceil$. The values $M(i, \ell)$ can be computed by

solving the following recurrence equation:

$$\begin{aligned} M(0,\ell) &= 0 \text{ for all } \ell = 0,1,\ldots,\lceil 2n/\varepsilon\rceil,\\ M(i,\ell) &= \max\{M(i-1,\ell), M(i-1,\ell-r_i)+s(a_i)\},\\ &\quad \text{where } y_{s(a_i)}/n(s(a_i)) = r_i\varepsilon P/n, \text{ for all } i=1,\ldots,n \quad \text{and}\\ &\quad \ell = 1,\ldots,\lceil 2n/\varepsilon\rceil. \end{aligned}$$

A simple algorithm for solving this recurrence uses $O(n^2/\varepsilon)$ time. Once we have computed the values $M(i,\ell)$, we find the smallest price value $\ell'(\varepsilon/n)P$ for which there is a solution of size $M(n,\ell') \geqslant 1$.

Lemma 2. *An optimum solution for problem* (6) *has value at most*

$$(1+\varepsilon)\ \min\left\{ j\frac{\varepsilon}{n}P + \frac{y_{|H|+1}}{s(\mathscr{S})}\sigma(1-M(n,j)) \mid j=0,\ldots,\ell' \right\},$$

where $\sigma(x)=\max\{x,0\}$. *Thus, a* $(1+\varepsilon)$*-approximation for* (6) *can be computed in* $O(n^2/\varepsilon)$ *time.*

Proof. Consider problem (6) with coefficients $y_v/n(v)$ rounded as described above. To prove the lemma we just need to show that an optimum solution for the rounded problem has value

$$\min\left\{ j\frac{\varepsilon}{n}P + \frac{y_{|H|+1}}{s(\mathscr{S})}\sigma(1-M(n,j)) \mid j=0,\ldots,\ell' \right\}.$$

Let $\hat{\tau}_1$ and $\hat{\tau}_2$ be optimum solutions for the rounded Problems (4) and (5), respectively. Let $\hat{\tau}=\min\{\hat{\tau}_1,\hat{\tau}_2\}$, and $\hat{z}$ be a solution of value $\hat{\tau}$. If $\hat{\tau}=\hat{\tau}_2$, the set $T\subseteq L$ used to compute the value $M(n,\ell')$ is the minimum price set of size at least 1, and so $\sum_{a_i\in T} y_{s(a_i)}/n(s(a_i)) = \ell'(\varepsilon/n)P = \hat{\tau}_2$.

On the other hand, if $\hat{\tau}=\hat{\tau}_1$, let $\hat{T}$ be the set of items in an optimum solution $\hat{z}$. Let $\sum_{a_i\in\hat{T}} y_{s(a_i)}/n(s(a_i)) = \hat{r}(\varepsilon/n)P$. Then, by definition of $M(j,\ell)$, $s(\hat{T}) \leqslant M(n,\hat{r})$. Since $1-s(\hat{T}) \geqslant 0$ then $\hat{r}\leqslant\ell'$, and so $\hat{\tau}_1 = \hat{r}(\varepsilon/n)P + y_{|H|+1}/s(\mathscr{S})(1-s(\hat{T})) \geqslant \hat{r}(\varepsilon/n)P + y_{|H|+1}/s(\mathscr{S})(1-M(n,\hat{r})) \geqslant \min\{ j(\varepsilon/n)P + y_{|H|+1}/s(\mathscr{S})\sigma(1-M(n,j)) \mid j=0,\ldots,\ell'\}$. □

Theorem 1. *There is an algorithm A that, given a set L of n items with sizes* $s(a_i)\in(0,1)$ *and a positive number* $\varepsilon>0$, *produces a bin covering of L such that* $A(L)\geqslant (1-\varepsilon)OPT(L) - O(1/\varepsilon^3)$. *The time complexity of A is polynomial in n and* $1/\varepsilon$.

Proof. The time needed for approximately solving linear program (1) is $O(|H|(1/\varepsilon^2 + \ln|H|)\ln(n/\varepsilon)\max\{n^2/\varepsilon, |H|\ln\ln(|H|/\varepsilon)\})$. This solution is modified as described in Section 3 to get a basic feasible solution. This post-processing can be performed in $O((1/\varepsilon^4)\mathscr{M}(1/\varepsilon^2))$ time, where $\mathscr{M}(n)$ is the time to invert an $(n\times n)$ matrix. Since $|H| = O(1/\varepsilon^2)$, the overall running time is $O((1/\varepsilon^5)\ln(n/\varepsilon)\max\{n^2, (1/\varepsilon)\ln\ln(1/\varepsilon^3)\} + (1/\varepsilon^4)\mathscr{M}(1/\varepsilon^2))$.

By proceeding similarly as in [4] one can show that if $\tilde{s}(L) > 13/\varepsilon^3$, the total number of bins covered by our solution is at least $(1-\varepsilon)OPT(L) - 1$, and if $\tilde{s}(L) \leqslant 13/\varepsilon^3$ our algorithm covers $OPT(L) - O(1/\varepsilon^3)$ bins. □

References

[1] S.B. Assman, Problems in discrete applied mathematics, Ph.D. Thesis, Department of Mathematics, MIT, Cambridge, MA, 1983.
[2] S.B. Assman, D.S. Johnson, D.J. Kleitman, J.Y-T. Leung, On the dual of the one-dimensional bin packing problem, J. Algorithms 5 (1984) 502–525.
[3] J. Csirik, J.B.G. Frenk, M. Labbe, S. Zhang, Two simple algorithms for bin covering, Acta Cybernet. 14 (1999) 13–25.
[4] J. Csirik, D.S. Johnson, C. Kenyon, Better approximation algorithms for bin covering, Proc. of SIAM Conf. on Discrete Algorithms, Washington, DC, 2001, pp. 557–566.
[5] M.D. Grigoriadis, L.G. Khachiyan, Coordination complexity of parallel price-directive decomposition, Math. Oper. Res. 21 (1996) 321–340.
[6] K. Jansen, H. Zhang, Approximate algorithms for general packing problems with modified logarithmic potential function, Proc. 2nd Internat. Conf. on Theoretical Computer Science, Montreal, Canada, 2002, pp. 255–266.